NPS-OR-00-005

NAVAL POSTGRADUATE SCHOOL
Monterey, California

An Integer-Linear Program to Plan Procurement and Deployment of Space and Missile Assets

by

Alexandra M. Newman
Gerald G. Brown
Robert F. Dell
Captain Angela Giddings, USAF
Richard E. Rosenthal

April 2000

Approved for public release; distribution is unlimited.

Prepared for: AFMC Office of Aerospace Studies
Kirtland AFB, NM

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5000

RADM Robert C. Chaplin
Superintendent

Richard Elster
Provost

This report was prepared for and funded by the Air Force Materiel Command Office of Aerospace Studies, Kirtland AFB, NM.

Reproduction of all or part of this report is authorized.

This report was prepared by:

Alexandra M. Newman

ALEXANDRA M. NEWMAN
Research Assistant Professor of
Operations Research

GERALD G. BROWN
Professor of Operations Research

ROBERT F. DELL
Associate Professor of
Operations Research

Alexandra M. Newman for

ANGELA GIDDINGS
Captain, USAF

Richard E. Rosenthal

RICHARD E. ROSENTHAL
Professor of Operations Research

Reviewed by:

R. KEVIN WOOD
Associate Chairman for Research
Department of Operations Research

Richard E. Rosenthal

RICHARD E. ROSENTHAL
Chairman
Department of Operations Research

Released by:

DAVID W. NETZER
Associate Provost and Dean of Research

REPORT DOCUMENTATION PAGE

Form approved

OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 2000	Technical

4. TITLE AND SUBTITLE

An Integer-Linear Program to Plan Procurement and Deployment of Space and Missile Assets

5. FUNDING

N0003900WRDR036

6. AUTHOR(S)

Alexandra M. Newman, Gerald G. Brown, Robert F. Dell,
Captain Angela Giddings, USAF, and Richard E. Rosenthal

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

NPS-OR-00-005

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFMC Office of Aerospace Studies
Kirtland AFB, NM

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Space Command Optimizer of Utility Toolkit (SCOUT) is an integer linear program used by the Air Force Office of Aerospace Studies to help plan the research and development of space-based systems over a 25-year horizon. SCOUT recommends a mix of concepts, current systems, and launches that minimizes shortfalls in task performance while adhering to constraints on budget, launcher demand, launcher availability, and logic governing the precedence and interdependence of systems. This technical report provides a mathematical description of SCOUT, details several modifications, and reports computational experience. We find a significant reduction in solution time by using discount factors, and introducing the notion of continuous-valued *research and development concepts*.

14. SUBJECT TERMS

capital budgeting, integer linear programming

15. NUMBER OF PAGES

36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

An Integer-Linear Program to Plan Procurement and Deployment of Space and Missile Assets

Alexandra M. Newman
Gerald G. Brown
Robert F. Dell
Captain Angela Giddings, USAF*
Richard E. Rosenthal

Operations Research Department
Naval Postgraduate School
Monterey, CA 93943

*Air Force Materiel Command Office of Aerospace Studies
Kirtland AFB, NM

12 APRIL 2000

ABSTRACT

The Space Command Optimizer of Utility Toolkit (SCOUT) is a mixed-integer linear program used by the U.S. Air Force Office of Aerospace Studies to help create a 25-year investment plan for space-related systems (e.g., space-based radars and space planes). SCOUT recommends a mix of current systems, "future concepts," and launches that minimizes shortfalls in task performance while adhering to constraints on budget, launch vehicle demand, launch vehicle availability, and logic governing the precedence and interdependence of systems. This technical report provides a mathematical description of SCOUT, describes several potential modifications to SCOUT, and reports computational experience. Our results suggest that SCOUT's computational times can be significantly reduced (from hours to minutes) and SCOUT's integrality gaps can be tightened by applying discount factors to both costs and tasking shortfall penalties, and by using continuous variables to model some research-and-development concepts.

I. INTRODUCTION AND BACKGROUND

The U.S. Air Force Space Command's operational tasks include: supporting space forces (e.g., launching and operating U.S. military satellites), tracking objects orbiting the earth, disseminating relevant information to warfighters (e.g., weather updates and missile warnings), and maintaining a combat-ready intercontinental ballistic missile force. The Space Command Optimizer of Utility Toolkit (SCOUT) is a mixed-integer linear program that selects from a pool of *candidate systems* a set of assets such as launch vehicles, satellites, and radars (see Figure 1), the dates of inception and discontinuance of use of these systems, and the number of launches (see Figure 2) by type and year that best satisfy Air Force Space Command's operational tasks over a 25-year horizon. (The appendix provides a list of tasks.) SCOUT adheres to constraints on budget, launch vehicle demand, launch vehicle availability, and logic governing the precedence and interdependence of systems. In 1997, Air Force Space Command used a version of SCOUT as one of several tools in a biennial investment analysis. Rankings derived from the analysis helped secure monetary support for the Military Space Plane, the Space-Based Radar, the Space-Based Laser, GEODSS Upgrades, the Satellite Threat Warning and Attack Reporting System, the Common Aero Vehicle, and Small Aperture Telescope Augmentation [Gooley 1998]. This technical report provides a mathematical description of SCOUT, details several modifications to SCOUT to reduce solution time and enhance solution quality, and reports computational experience.

SCOUT candidate systems consist of research-and-development concepts and pre-existing weapon and defense programs. Space-based radars, radar tactical satellites, terrestrial weather sensor upgrades, and advanced electro-optical warning sensors are all examples of systems. Most systems require launch vehicles of a specific type, ranging from ultralight to ultraheavy, where the heavier launch vehicles can be used to satisfy lighter launch requirements. We assume that any single launch can only lift a single system into its orbit.

Parnell, Conley, Jackson, Lehmkuhl and Andrew [1998] provide methods for quantifying a system's contribution towards task performance. SCOUT recognizes that systems contribute towards task performance in a non-additive manner. Each system is scored between 0% and 100% in 10% increments according to the approximate percent of coverage (or contribution) it alone provides towards a single task. The contribution from several systems that perform the same task is either the maximum contribution of any single system or the maximum "synergistic contribution" of two or more systems. For example, if system A contributes a 30% task-performance level and system B contributes 40% to performing the same task, the total

contribution to this specific task is only 40% performance without synergy. Two or more systems are synergistic if they together provide greater task performance than their maximum individual performance.

SCOUT constraints include annual budget restrictions, budget constraints over a contiguous set of years (i.e., an "epoch"), annual launch vehicle requirements, dependency and precedence relationships among systems, and "bookkeeping" relationships. SCOUT has two budget constraints; each constraint has an elastic allowance (subject to a linear penalty per unit of violation) for exceeding its specified limit. Annual budget constraints restrict all costs incurred after a system is fully operationally capable (FOC) except launch costs. The FOC year for a system is the first year that a system contributes to meeting tasking requirements. Additional budget constraints restrict all costs over five-year epochs. These constraints enhance SCOUT's realism because some space systems require that the bulk of their costs be spent on research, development and initial launch. These activities generally occur prior to the realization of operational benefits from the systems. The additional budget constraints allow for more latitude than the annual budget constraints to pay for launch vehicles. Per-unit launch costs are not subject to strict annual budget requirements, because launch vehicles may be inventoried and subsequently launched when needed. Without this allowance, solutions would be precluded that require a large number of launch vehicles in a single year.

SCOUT selects systems and system launches according to rules governing their compatibility and interdependence: (i) there must be sufficient launch capabilities to provide the required number of launches for a given system, launch vehicle type and year; (ii) a launch vehicle must be operational before it can be used to provide launches; (iii) systems dependent on a primary system can be operational only if the primary system is already operational; and (iv) prerequisite systems must be, or must have been, operational before a secondary system can become operational.

SCOUT minimizes total penalties, composed of contributions from: (i) a penalty associated with shortfall in task performance over time; (ii) a penalty associated with the violation of either annual or epochal budget constraints; and (iii) a small penalty to discourage the model from spending additional money if no additional task performance is gained. All penalties are chosen to provide objective-function units of 1998 U.S. base-year dollar equivalents.

In the following section, we present a review of previous literature on capital budgeting models used both in civilian and military arenas. Section 3 introduces a version of SCOUT developed by the Office of Aerospace Studies and the Naval Postgraduate School in 1997 and used by Air Force Space Command. Computational experience and an assessment of the quality

of solutions obtained from the suggested modifications to SCOUT are given in Sections 4 and 5, respectively. Our results suggest that SCOUT's computational times can be significantly reduced (from hours to minutes) and SCOUT's integrality gaps can be tightened by applying discount factors to both costs and tasking shortfall penalties, and by using continuous variables to model some research-and-development concepts; solution quality is not sacrificed with these modifications. Conclusions appear in Section 6.

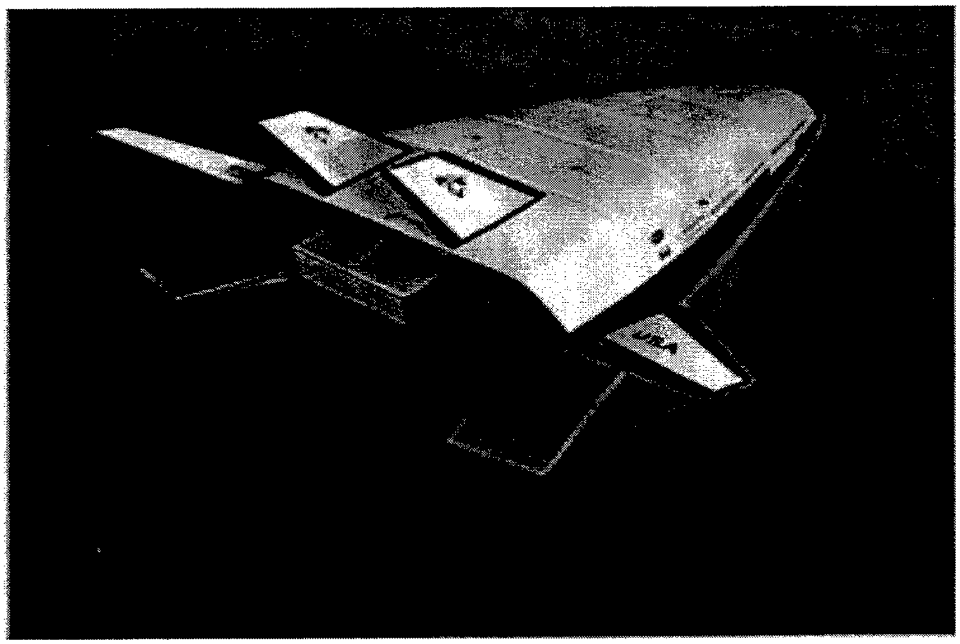

Figure 1: An artist's rendition of the X-33 Venture Star (space plane) [United States Air Force 1999b], one of the *concepts* the Air Force may fund to satisfy its future *operational tasks*. The development of concepts such as the space plane often requires funding for ten years or more before any benefit is realized. The U.S. Air Force used a version of SCOUT, a mixed-integer linear program, to help plan the procurement of billions of dollars worth of space-related systems. SCOUT selects a set of systems, the dates of inception and discontinuance of use, and the number of launches by type and year over a 25-year time horizon.

II. LITERATURE REVIEW

SCOUT is a capital-budgeting model in the taxonomy of optimization modeling applications (e.g., Clark, Hindelang, and Pritchard [1989] and Weingartner [1963]). There are hundreds of published references to capital-budgeting models; most of these examples differ from SCOUT in two significant ways: (i) few address military applications, and (ii) few authors concern themselves with real-world, large-scale applications with complete decision-support implementations—as such, most earlier work is somewhere between theoretical and hypothetical. We are concerned with actually solving models much larger than most of those in the literature. Accordingly, the following citations are selected to provide context for SCOUT but do not offer guidance to enhance SCOUT. (The reader may omit this section without loss of continuity with the remainder of the report.)

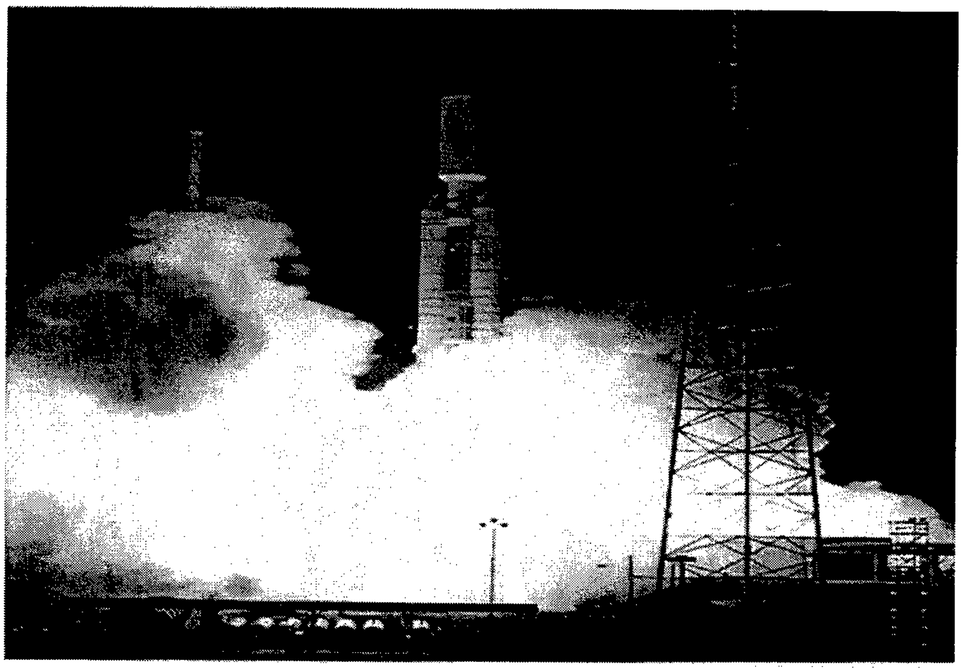

Figure 2: A Titan IV/Centaur, an expendable launch vehicle able to lift a 10,000 pound payload into orbit, launches from Cape Canaveral Air Station carrying a Milstar satellite [United States Air Force 1999a]. SCOUT recognizes five launch vehicle types (based on payload capabilities) to ensure adequate launch resources for a *dependent system* (in this case, a satellite). SCOUT models other dependencies such as systems requiring the (not necessarily concurrent) existence of other systems, and *synergistic relationships* enhancing the simultaneous performance of several systems operating in tandem.

In a methodological paper, Hummeltenberg [1985] proposes the use of Benders decomposition for solving capital-budgeting problems in which different interest rates are assessed for borrowing and lending. His model seeks to maximize available cash at the end of the planning horizon subject to balance of cash flow and borrowing constraints. Hummeltenberg

emphasizes the importance of the following factors: (i) problem formulation, (ii) the initial set of Benders cuts, (iii) choice of master-problem solutions, and (iv) cut selection if the subproblem has multiple optimal solutions. Extensive computational testing is performed on both investment and financing projects with a ten-year time horizon. These test problems are solved in a few seconds on a mainframe computer and are used to illustrate the relationship between parameter values, the specific implementation scheme, and computational tractability. Although Benders decomposition may be a viable approach for improving SCOUT's performance, we cannot infer this based on the unspecified size and structure of Hummeltenberg's models.

Several authors address capital budgeting for production and manufacturing. The following three references contain examples of problem instances smaller than SCOUT, which the respective authors claim are realistic for the scenarios offered. The contribution of the work lies in the managerial insights gained from the solutions, rather than from any advice on implementation.

Keown and Taylor [1980] minimize manufacturing costs while adhering to environmental standards, and they introduce chance constraints to account for uncertainty in demand. They employ goal programming to minimize deviation from the following conflicting objectives, in decreasing order of importance: (i) meet environmental restrictions, (ii) recognize inventory storage capabilities, (iii) comply with budgets, (iv) minimize excess capacity, (v) contribute to profit, (vi) increase sales, (vii) satisfy demand, and (viii) open sufficient facilities to perform daily operations. Space availability for new warehouses is enforced as a hard constraint. A small example containing fewer than 20 constraints and 35 variables is solved.

Kumar and Lu [1991] present a case study for a real-world capital-budgeting problem for a fertilizer production facility. They develop a mixed-integer linear program to account for economies of scale and system interdependencies. The model seeks to maximize profit subject to: (i) conservation of materials flow, (ii) budget adherence, (iii) demand satisfaction, (iv) raw-material availability, (v) plant capacity based on both physical and legal limitations, and (vi) interdependencies in decision making, e.g., the necessity to produce precursor chemicals. Their problem instance contains about 30 constraints and 70 variables; only about ten variables are binary. Data perturbation proves useful as a post-optimization tool.

Lotfi, Sarkis and Semple [1998] apply a strategic capital-budgeting model to determine the benefit of replacing a traditional manufacturing system with an advanced "flexible manufacturing system." The goal of the model is to maximize the net present worth of a manufacturing system subject to restrictions on capacity, requirements that demand be met, limits on the number of production systems that can be operational simultaneously, and requirements on

the continuity of operation (specifically, if a system becomes operational at some point in the planning horizon, it remains operational throughout that horizon). They transform their integer program into a model whose linear-programming relaxation yields integer solutions. Problem instances are generated that contain 24 variables and 10 constraints under the transformed formulation, and can be solved easily.

Iwamura and Liu [1998] address a capital-budgeting model of production in order to demonstrate a new modeling approach. Similar to Keown and Taylor [1980], they employ chance constraints in a fuzzy, rather than stochastic, environment. Their model seeks to minimize deviation from (i) budget limits, (ii) physical space limits, and (iii) a profit goal. They develop a "fuzzy" simulation-based genetic algorithm to solve very small, randomly generated problems using a personal computer.

A variety of literature applies capital-budgeting models to areas other than production. Again, these models are smaller than SCOUT, and the setting and nature of the decisions differ. Mamer and Shogan [1987] consider a capital-budgeting model applied to repair kit selection that requires choosing projects along with an associated set of activities; individual activities can be used to satisfy the requirements for more than one project. A fixed cost and resource utilization are associated with each activity. The formulation is a maximum-flow network model with a resource side constraint. Problem instances containing up to several hundred constraints and 600 variables, a third of which are binary, are generated and solved via Lagrangian relaxation in less than three minutes on an IBM mainframe computer. The resulting optimality gaps (the difference between the best solution and the best bound) range from 1% to 30%, indicating that this solution technique is not reliable for all problem instances.

Karabakal, Lohmann and Bean [1994] formulate a model for parallel replacement of resources subject to rationing constraints. Parallel replacement differs from serial replacement in that the former requires investment decisions to be made in each period of the planning horizon to adequately capture economic interdependencies between assets. The model seeks to maximize total net present value subject to conservation of flow and capital rationing (budget constraints). This model can also be viewed as a network model with side constraints. Randomly generated problems for a four- to ten-period time horizon containing no more than 120 constraints and 1,000 binary variables are solved using a Lagrangian branch-and-bound algorithm on an IBM mainframe computer in generally less than a minute, although several run times of 30 minutes to about three hours are reported. Problems are solved to near-optimality in many, though not all, instances.

Capital budgeting has been modeled for the military. Taylor, Keown and Greenwood [1983] address the procurement of military aircraft to satisfy military objectives in wartime and peacetime while minimizing violation of the following elastic constraints: (i) interdependence among weapon systems and the use of aircraft, (ii) ability to achieve a desired number of kills, (iii) budget limitations, (iv) aircraft payload and range, (v) aircraft maintenance feasibility, (vi) peacetime aircraft loss, and (vii) flexibility (interoperability) of systems with other allied systems. A small mixed-integer program containing approximately 25 variables and 25 constraints is solved with a branch-and-bound algorithm.

Brown, Clemence, Teufert and Wood [1991] develop a large-scale, real-world model for long-term planning to modernize the U.S. Army's helicopter fleet by taking the following actions: (i) manufacturing aircraft via completely new production plans, (ii) modifying existing production campaigns to incorporate aircraft enhancements, (iii) extending the life of existing aircraft through upgrading, and (iv) retiring outdated aircraft. The model seeks to minimize operating and maintenance costs subject to limits on levels of performance, average age, expenditure, production and manufacturing requirements, and factors necessary for the conversion of aircraft. Research-and-development cost streams are similar to those in SCOUT, i.e., initial costs are incurred before an investment becomes fully operationally capable. An actual scenario with a 25-year time horizon, 16 helicopter types, 300 potential production campaigns, and five production lines is reported. The model contains about 4,000 constraints and 21,000 variables, 300 of which are binary. Most problem instances are solved to optimality in five to ten minutes on an IBM mainframe. Solutions reveal that the aircraft fleet had to be reduced, resulting in near-term mission deficiencies, and that significant benefits accrued when funding levels over the planning horizon were allowed to be non-uniform. That is, much better outcomes resulted from rescheduling expenditures but not changing their total. This insight led to restricting budget plans in terms of total program costs governed by annual targets to smooth, but not completely restrict, the pattern of expenditures over time. Contrary to preconceived notions, certain nearly-new aircraft were not cost effective. Furthermore, promising alternatives to the then-current planning *modus operandi* necessitated violations of other resource and policy guidelines. The model has been credited with changes the Army adopted and approvals by Congress and the White House. The work of Brown, Clemence, Teufert, and Wood provides our best example of a model both commensurate in size and related in application to SCOUT. However, the decisions and constraints differ, and we cannot be certain of such good computational performance with SCOUT.

Capital-budgeting models have been formulated and solved with actual data in a variety of Masters theses at the Naval Postgraduate School. These theses provide excellent examples of real models; however, they tend to be smaller than SCOUT, and special implementation techniques for improved performance are not offered. Donahue [1992] examines the question of which Army projects to fund in long-term requirements planning. The model is a multi-criterion optimization model. Continuous variables are used to represent the fraction of aspired funding a proposed project (analogous to a concept in SCOUT) will be allocated in a given year. Binary variables are used to represent logical relationships between projects. The objective is primarily to obtain a desired level of fighting capability while secondarily preserving balance across all areas of the force. Constraints are imposed to enforce: (i) the funding of government-mandated projects and projects requiring minimum expenditures, (ii) limits on budget and operational costs, (iii) precedence rules for project stages, (iv) relationships between discrete and continuous variables, and (v) required relationships among mutually exclusive, synergistic, and subordinate projects. We note that the constraints on initiating synergistic and subordinate projects are similar in spirit to the synergy and precedence constraints in SCOUT. An instance of the model, which contains over 5,000 variables, 230 of which are binary, and about 5,000 constraints, is solved on a personal computer in a few seconds. The model has been used both by the Special Operations Command to assemble their Program Objective Memorandum and by the Army Training and Doctrine Command to assist in research, development and acquisition expenditure planning [Anderson 1999].

Ihde [1995] develops a model for determining a maximally effective purchasing schedule for anti-armor warfare equipment subject to budget limits and selection restrictions. He considers both positive and negative effectiveness correlations between pairs of weapons. By changing the budget constraints and using different measures of effectiveness, different purchasing plans are realized. His model contains about 600 constraints and an equal number of variables, of which almost 100 are binary. Most problem instances are solved in less than two minutes on a personal computer. Carr [1996] addresses the procurement of theater missile-defense systems subject to constraints on budget, production, logical relationships concerning the constitution of a complete weapons system, and operational requirements. He evaluates different scenarios resulting from a modification of budget restrictions, weapon systems' operational requirements and associated procurement policies, cost parameters, and production rates. This model contains about 1,800 constraints and approximately 950 variables, of which 24 are binary. Most instances are solved in less than two minutes with a personal computer.

Gross [1996] develops two models to maximize the effectiveness of weapons systems using a given procurement schedule. Both models contain constraints on budget, minimum effectiveness, production limits, and the relationship between options and weapons systems. Tradeoffs are analyzed among budget levels, effectiveness measures, weapons systems, and different regions of conflict. The simpler model contains about 1,400 constraints and 650 variables, of which approximately half are binary. Instances are solved on a personal computer in a few minutes. The more complicated formulation, allowing more detailed system effectiveness correlation, contains over 22,000 constraints and almost 16,000 variables, almost all of which are restricted to be binary. These problem instances are solved on an IBM RS6000 Model 590 workstation in about one hour.

In solving many large-scale applications such as the military models mentioned above, it is common to elasticize constraints. This allows a greater set of solutions to be considered and can increase model tractability without sacrificing its initial intentions. For example, budget limits are often relaxed by adding a variable to account for over- or under-spending. A "high" linear penalty is applied to this term to discourage spending beyond the original limits. Other inequality constraints expressing limits or desired goals may be relaxed in the same spirit. Brown, Clemence, Teufert, and Wood [1991], and Ihde [1995], among others, provide illustrations of models with elastic constraints. Brown, Dell, and Wood [1997] cite additional real-world applications containing elastic constraints.

III. DEFINITIONS AND FORMULATION FOR SCOUT

We now present SCOUT adhering (as closely as possible) to the notation adopted for the Office of Aerospace Studies' initial implementation. Units for data and decision variables are given in brackets next to the respective definition. It is understood that compound index sets are only defined for the instances that exist.

A. INDICES

(i) Tasks and options

i	tasks
j, j'	systems, concepts and current programs (e.g., space-based radars, radar tactical satellites, terrestrial weather sensor upgrades); we use j' exclusively to denote systems on which other systems j depend
k	increasing levels of task performance (1-10)
l, l'	launch vehicle types (e.g., ultralight, light, medium, heavy, ultraheavy)

(ii) Time

t, t', t''	fiscal year index (e.g., $t = 1$ (1998), 2 (1999),..., 25 (2022)); we use t' (t'') as the start year index when paired with index j (j')
s, s'	end year index for a type j system; and end year index for a type j' system, respectively
p	epoch (contiguous set of years)

B. SETS

(i) Years

T_p	set of contiguous years t in epoch p

(ii) Performance levels

Dv_{it}	set of attainable performance levels k of task i in year t

Mx_{ik}	set of maximum performance levels k for task i with systems of type j (this set contains as many elements as there are systems that can meet a given tasking requirement i)

(iii) Systems

Lr_l	set of systems j requiring a launch vehicle of type l or greater
Ls_l	set of systems j providing launches of type l
Or	set of systems j
Ift	set of system pairs (j, j') in which system j can only be used while system j' is operational
Pre	set of system pairs (j, j') in which system j cannot become operational unless system j' has been developed to operational capability (differs from $Ift(j,j')$ in that system j may continue to operate after j' ends operation)
Syn	set of "synergistic" system pairs (j, j')
$Tv_{t's}$	set of systems j that begin operation in year t' and end in year s (may include dependent systems; start and stop years refer to the operational lifetime of a system, which does not necessarily match its spending lifetime)
$Tv'_{t''s'}$	set of systems j' that begin operation in year t'' and end in year s' but includes only systems on which other systems depend
Ω_t	set of systems j that begin operation in year t' and end in year s with non-zero cost in year t, where $j \in Or \cap Tv_{t's}$, $t' \le t + TIMETFOC_j$, and $s \ge t'$.

C. DATA

(i) Time intervals

$TIMETFOC_j$	number of years of non-zero costs for system j until the year prior to FOC [years]
$INTERVAL$	length of interval which budget constraints span [years]

(ii) Objective function weights

BIGM	weight for task performance shortfall [millions of 1998 penalty dollars per million 1998 US dollars]
BUDGM	objective function weight for total expenditures [millions of 1998 penalty dollars per million 1998 US dollars]
SMALLM	objective function weight for budget violations [millions of 1998 penalty dollars per million 1998 US dollars]

(iii) Budget limits

$BUDG_t$	budget allocated for year t [millions of 1998 US dollars]
$BUDG_p$	budget allocated for epoch p [millions of 1998 US dollars]

(iv) Costs

$COST_{jt(t\text{-}t'+TIMETFOC_j+1)}$	cost to develop system j incurred in year t for the $(t\text{-}t'+TIMETFOC_j+1)^{th}$ year of non-zero costs; this cost excludes launches, and $t'\text{-}TIMETFOC_j \leq t$ [millions of 1998 US dollars]
$UNITCOST_{jt}$	cost per launch in year t for a launch system j [millions of 1998 US dollars per launch]

(v) Elastic penalties

PEN_{it}	penalty per level of task i performance shortfall in year t [millions of 1998 US dollars per unit shortfall in task performance]

(vi) Launch requirements

$BIGMLR_{jlt}$	maximum number of system j launches for launch type l in year t [launch]
$LNCHREQ_{jt}$	number of required launches for system j in year t (where for a system j, an eligible year t is referenced from the first FOC year, t', and $t'\text{-}TIMETFOC_j \leq t$) [launch]

D. VARIABLES

(i) Continuous, nonnegative variables

$bover_t$	amount by which annual budget constraints are exceeded in year t [millions of 1998 US dollars]
$lover_p$	amount by which epochal budget constraints are exceeded in epoch p [millions of 1998 US dollars]
$xdev_{it}$	shortfall in meeting task i in year t [unit shortfall in task performance, i.e., 1, 2, 3,...]
xn_{jlt}	number of system j launches in year t with a launch vehicle of class l [launch]

(ii) Binary variables

$$x_{jt's} = \begin{cases} 1 & \text{if system } j \text{ is started in year } t' \text{ and operated until year } s \\ 0 & \text{otherwise} \end{cases}$$

E. FORMULATION

$$\min z = BIGM * \sum_{i,t} PEN_{it} * xdev_{it} + \tag{1a}$$

$$+ SMALLM * \left(\sum_{p=1}^{\lceil 25/\mathrm{INTERVAL} \rceil} lover_p + \sum_{t=1}^{25} bover_t \right) \tag{1b}$$

$$+ BUDGM * \left(\sum_j \sum_l \sum_t UNITCOST_{jt} * xn_{jlt} + \sum_t \sum_{j,t',s\in\Omega_t} COST_{jt(t-t'+TIMETFOC_j+1)} * x_{jt's} \right) \tag{1c}$$

subject to

$$\sum_{j,t',s\in\Omega_t} COST_{jt(t-t'+TIMETFOC_j+1)} * x_{jt's} \le BUDG_t + bover_t \qquad \forall t \tag{2}$$

$$\sum_{t \in T_p}\left(\sum_{j,l} UNITCOST_{jt} * xn_{jlt}\right) + \sum_{t \in T_p}\sum_{j,t',s \in \Omega_t} COST_{jt(t-t'+TIMETFOC_j+1)} * x_{jt's} \le BUDG_p + lover_p \qquad \forall p \tag{3}$$

$$\sum_{t' \le t}\sum_{s \ge t}\sum_{l' \ge l}\sum_{j \in Lr_{l'} \cap Tv_{t's}} LNCHREQ_{j(t-t'+TIMETFOC_j+1)} * x_{jt's} \le \sum_{l' \ge l}\sum_{j \in Ls_{l'}} xn_{jl't} \qquad \forall l,t \tag{4}$$

$$xn_{jlt} \le \sum_{t' \le t}\sum_{s \ge t} BIGMLR_{jlt} * x_{jt's} \qquad \forall t,l,j \in Ls_l \tag{5}$$

$$x_{jt's} \le \sum_{t'' \le t'}\sum_{s' \ge s} x_{j't''s'} \qquad \forall t',s \quad \forall (j,j') \in Syn \cup Ift, j \in Tv_{t's}, j' \in Tv'_{t''s'} \tag{6}$$

$$x_{jt's} \le \sum_{t'' \le t'} x_{j't''s'} \qquad \forall t',s,s' \quad \forall (j,j') \in \Pr e, j \in Tv_{t's}, j' \in Tv'_{t''s'} \tag{7}$$

$$xdev_{it} + \sum_{t' \le t}\sum_{s \ge t}\sum_{j \in Mx_{ik}} [\max\{k\} - k + 1]\, x_{jt's} \ge \max\{k\} - k + 1 \qquad \forall i,t,k \in Dv_{it} \tag{8}$$

$$bover_t \ge 0 \ \ \forall t, \quad lover_p \ge 0 \ \ \forall p, \quad xdev_{it} \ge 0 \ \ \forall i,t, \quad xn_{jlt} \ge 0 \ \ \forall j,l,t, \quad x_{jt's} \ge 0 \text{ and binary } \forall j,t',s \tag{9}$$

The objective function minimizes total 1998-penalty dollars, comprising contributions from: (1a) the penalty associated with annual shortfall in task performance, (1b) the penalty associated with the violation of either annual or epochal budget constraints, and (1c) a small penalty to discourage spending if no additional task performance is gained.

Constraints (2) are annual elastic budget limits for expenditures excluding launch costs. SCOUT uses a system's FOC year and the years of non-zero funding before FOC (i.e., *TIMETFOC*) as references for the system's cost stream. For example, suppose system j must incur $TIMETFOC_j = 3$ years of non-zero funding before it is started in year $t' = 5$. In year $t = 4$, the cost incurred is $COST_{j3}$, where 3 corresponds to the third year of non-zero funding incurred by the system. Constraints (3) elastically enforce budget constraints including launch costs for each epoch in the planning horizon, where an epoch has a total length of *INTERVAL* years. The costs consist not only of those costs incurred before the system is fully operational, as in constraints (2), but also of annual per-launch costs.

Constraints (4) require that the total number of launches provided in each launch class in a given year t must meet or exceed the total number required in that year. As in the previous two constraints, time for the required number of launches is measured with respect to the first year of non-zero costs for the system. The indexing, therefore, parallels that of the *COST* parameters in the previous two constraints. Constraints (5) require that a launch system j be operational in a given year t before it provides launches (of class l) in that year. Constraints (6) allow dependent systems j to be operational in a given year only if all of the systems j' they require are operational in that year. Constraints (7) ensure that precedence relationships hold across time. A precedence relationship requires system j' be operational before system j can become operational. However, the two systems are not required to be operational simultaneously. Constraints (8) determine the annual task shortfalls. Constraints (9) enforce nonnegativity and integrality requirements.

SCOUT is implemented in the General Algebraic Modeling System (GAMS) [Brooke, Kendrick, Meeraus, and Raman 1997] with the CPLEX solver, Version 5.0 [ILOG 1998]. The model runs under an AIX operating environment on an IBM RS6000 Model 590 workstation with 0.5 gigabytes of RAM. After the CPLEX presolve (during which the model is simplified by, for example, eliminating redundant rows and columns), the model contains approximately ten thousand constraints and five thousand variables; about half of the variables are binary.

IV. COMPUTATIONAL RESULTS

We explore computational performance resulting from several SCOUT modifications. Our goal is to examine different versions of SCOUT that maintain the model's fidelity and show promise in reducing both computational requirements and *relative integrality gaps*. (The *gap* is the normalized difference between the cost of the best solution found (BI) and a lower bound (LB) on the best possible solution. Specifically, for a non-zero lower bound, the gap is defined as 100%*[BI-LB]/LB.) We explain the modifications to SCOUT, and then compare computational results for the original model and the models with modifications. The following chapter gives an analysis of the effects of selected modifications on solution quality.

A. MODIFICATIONS

We employ a discount factor to reflect the facts that: (i) fixed and variable costs must be expressed in constant-year dollars, and (ii) because the future is uncertain, it is more important to fulfill performance requirements in the near term than at the end of the planning horizon. Employing the discount factor helps to differentiate between present and future spending, and the relative importance of meeting task shortfalls in the near term and the far term, respectively, and thereby enhances the solver's ability to distinguish among admissible solutions. We modify the parameters, $UNITCOST_{jt}$, $COST_{jt(t-t'+TIMETFOC_j+1)}$, and PEN_{it}, to incorporate the calendar year t in which the cost is incurred; this ensures that all costs are expressed in constant-year dollars. We do not modify the $BUDG_t$ or $BUDG_p$ parameters.

To attempt to guide the branch-and-bound algorithm in a logical manner, we specify a branching hierarchy, i.e., an order in which the binary decisions (of whether to implement a system for a given time interval) are evaluated. We consider the union of systems providing synergistic support, or necessary for a subordinate system to be brought on line, regardless of the time interval. In three different trials, i.e., test runs, we establish two, three and four priority levels within the branching hierarchy for these independent systems, giving higher priority to those systems with more dependents. This yields a methodical way for the search procedure to progress, i.e., by making decisions about certain "principal" systems before making decisions concerning dependent systems.

Currently, systems can be introduced each year for the first five years of the planning horizon, and only in odd calendar years thereafter. (This reduces allowable start-year and end-year combinations.) This change was made from an earlier version in which systems could be brought on line during any year of the planning horizon. We extend this idea to further decrease allowable start-year and stop-year combinations. Specifically, we retain the one-year time

intervals for the first four years, and the two-year intervals for the subsequent six years. After the tenth year (i.e., 2007), we allow systems to be introduced every three years for the next nine years. Systems can last be introduced in the twenty-third year of the horizon, i.e., year 2020. Although this is a restriction of the original problem, it realistically reflects the greater need to be precise with timing in the immediate future rather than in the distant future.

Each binary variable represents a decision such as whether to invest in a launch vehicle or not: One cannot obtain half a launch capability with half a launch vehicle. However, for variables representing research-and-development initiatives, it is reasonable to consider scenarios in which an initiative may be partially funded, or may be completely funded, but not necessarily continuously over the time horizon. Because we cannot explicitly identify potential research-and-development concepts from the data, we nominate those variables that do not provide synergistic, prerequisite or simultaneous support to another system, and neither require a launch vehicle nor provide one. We allow these variables to be continuous-valued, which significantly reduces the number of binary variables in the model. We can embellish this modification by requiring each fractionally-funded system to be fully-funded by the end of the horizon. Let *Rad* be the set of research-and-development concepts, let $xr_{jt's}$ be the continuous decision variables corresponding to starting a concept j in year t' and operating it until year s, and let xd_j be the binary decision of whether concept j is ever brought on-line during the time horizon. Then, additional constraints are:

$$xr_{jt's} \leq xd_j \qquad \forall t', s \qquad \forall j \in Tv_{t's} \cap Rad \qquad \text{and}$$

$$\sum_{t'} \sum_{s \geq t'} xr_{jt's} = xd_j \qquad \forall j \in Tv_{t's} \cap Rad.$$

We implement our computational trials for two test cases: (i) Test Case 1 represents an original SCOUT model (September 1998) using an updated (January 1999) set of eligible systems, start, and end dates, and (ii) Test Case 2 represents a newer (January 1999) SCOUT version with updated (January 1999) data. We use X.1 and X.2, respectively, to represent the results, where X denotes a trial number. We use CPU time and the relative integrality gap as performance gauges. We use relative integrality tolerances of 5% and 10%, as noted, as stopping criteria.

We now summarize our computational experiments. In the next section, we provide numerical results based on these trials. The initial run is made without any modifications. In Trials 2.1, 2.2, and all subsequent (X.1 and X.2) trials, we use a 2% "penalty degradation rate" for deviation from a shortfall in task performance, and we adjust constant-year dollar cost parameters

$UNITCOST_{jt}$, $COST_{jt(t-t'+TIMETFOC_j+1)}$ so that cost grows at a rate of 2.5% annually (i.e., after adjusting for the time value of money, a system is 2.5% more expensive one year in the future). Trials 3a.1 (and 3a.2), 3b.1, and 3c.1 reflect the results of discounting *and* priority branching with four, three, and two categories, respectively. (We also implemented a branching hierarchy in which priority was given to the xd_j variables over the $x_{jt's}$ variables, and in which the opposite priorities were assigned; no improvement in performance was realized, so we do not report the results from these trials.) Trials 4.1 and 4.2 demonstrate the effect of using discounting *and* reducing the number of eligible start-year and stop-year combinations for a system. In Trials 5.1 and 5.2, we do not require research-and-development initiatives to be fully funded by the end of the planning horizon. Trials 5a.1 and 5a.2 reflect the additional requirement that all selected research-and-development initiatives be completely funded, but not necessarily continuously by the end of the time horizon. Table 1 displays the characteristics of each trial.

Trial	Characteristics
1	original model
2	cost and penalty discounting
3a	cost and penalty discounting and priority branching with four categories
3b	cost and penalty discounting and priority branching with three categories
3c	cost and penalty discounting and priority branching with two categories
4	cost and penalty discounting and a reduced number of start-year and stop-year combinations
5	cost and penalty discounting and continuous-valued research-and-development variables
5a	cost and penalty discounting and continuous-valued research-and-development variables where chosen initiatives must be fully funded by end of horizon
6	combination of Trials 3 and 4
7	combination of Trials 4 and 5
8	combination of Trials 3, 4, and 5

Table 1: We modify the original model (Trial 1) in Trials 2-8 and use X.1 and X.2 to respectively represent the results, where X denotes the trial number and the suffix denotes the data set.

B. RESULTS

The original instances for both test data sets (i.e., Trials 1.1 and 1.2) yield about a 19% relative integrality gap after 5,000 seconds. In the first case, after nearly two hours, the gap is reduced below the tolerance goal of 10%, whereas in the second case, computer memory limitations prevent an improvement. A more stringent relative integrality tolerance of 5% is not satisfied in either case. We report results for both test cases in Tables 2 and 3 for the original

instances and the modifications using different required tolerances (i.e., 5% and 10%) as stopping criteria.

Performance is much improved with the addition of discounting (Trials 2.1 and 2.2). The gap is substantially reduced in both instances, though the run time for Trial 2.1 is far less than that for Trial 2.2. The priority branching scheme with fewest categories yields the best performance in Test Case 1; however, the result is slightly worse than that obtained without priority branching. Priority branching works well in Test Case 2; a solution within 10% of the optimal is obtained in one-fifth of the CPU time. Results for Trial 4.1 are not quite as impressive as those obtained in Trial 2.1. For Test Case 2, the run time for Trial 4.2 is lower than in Trials 2.2 and 3.2 for a solution guaranteed to be within 5% of the optimal; however, the run time for a solution satisfying the less stringent tolerance of 10% requires substantially more time to obtain. Trials 5.1 and 5.2 provide better results than Trials 2.1-4.1 and 2.2-4.2, respectively: in less than ten minutes of CPU time, we obtain solutions within 5% of optimal.

We add three trials to the second test case combining some modifications (see Table 1). In addition to the discount factors, Trial 6.2 combines priority branching and the change in time-period granularity. Trial 7.2 combines the change in time-period granularity and the introduction of continuous-valued research-and-development concepts. Finally, Trial 8.2 combines all three of these modifications.

requested gap **Trial**	**10%** *time (seconds)*	**5%** *time (seconds)*
1.1 (original)	6,862	74,909*
2.1	489	482
3a.1	704	702
3b.1	552	552
3c.1	1,067	1,067
4.1	464	4,355
5.1	305	305
5a.1	436	696

Table 2: Computational results for the original (September, 1998) SCOUT model when requiring a solution guaranteed to be within 10% and 5% of optimal. For example, Trial 2.1 requires approximately eight minutes (482 seconds) to find a solution within 5% of the optimal. In Trials 2.1 through 5a.1 (when requiring a solution within 10% of the optimal), a relative gap half as large, on average, can be achieved in about one-fourteenth of the time required for the original model. *This run was terminated (before achieving requested gap) due to memory limitations.

In general, for both test cases, the modifications yield solutions with less than a 10% integrality gap in a few minutes. Solutions with the modified formulation are not only obtained more quickly, they yield lower relative integrality gaps. Trials 4.2, 5a.2, and 8.2 are the only exceptions to this performance improvement; however, in the former two cases, although the run time is not particularly impressive when compared with the original scenario, the relative integrality gaps are much lower.

Moderation is a virtue when setting the relative integer tolerance. We observe that the solution obtained with this tolerance set at 10% is usually the best solution obtained within a reasonable amount of time. Trials 1.1, 2.2, 3a.2, 6.2, and 7.2 illustrate this point well. Trial 5a.1 provides the only strong counter-example to this argument. In actuality, the integrality gap must only be small enough to allow unambiguous comparisons of alternate, competing scenarios. The superiority of a scenario can be established if its best objective function value dominates the best that could conceivably be achieved with the competing version, regardless of the magnitude of the integrality gap for either scenario.

requested gap **Trial**	**10%** *time (seconds)*	**5%** *time (seconds)*
1.2 (original)	4,110*	—
2.2	3,375	25,051
3a.2	667	54,596*
4.2	6,553	6,553
5.2	576	581
5a.2	4,214	4,225
6.2	640	78,685*
7.2	497	5,269
8.2	11,260*	—

Table 3: Computational results for the January 1999 SCOUT version when requiring a solution guaranteed to be within 10% and 5% of the optimal. For example, Trial 2.2 requires approximately fifty-six minutes (3375 seconds) to find a solution within 10% of the optimal and almost seven hours (25,051 seconds) to obtain a solution within 5% of the optimal. In Trials 2.2 through 7.2 (when requiring a solution within 10% of the optimal), a relative gap about one-quarter as large, on average, can be achieved in less than 60% of the time required for the original model. *Runs terminated (before achieving requested gap) due to time or memory limitations; - data not reported due to excessive run time of the model.

V. QUALITY OF SOLUTIONS

The modifications we made to the model are designed to reduce the relative integrality gap and decrease solution time. Although we have, in general, accomplished these goals, we would like to assure ourselves that making modifications to the model, such as adding discount factors and introducing continuous-valued research-and-development variables, has not inherently changed the nature of the solutions from those obtained with the original models (i.e., September 1998 and January 1999). We focus our analysis on the solutions obtained from the second test case.

The first modification imposes a discount factor for the shortfall in task performance in a given year. We argue that performance shortfalls should be penalized more heavily in the near term because these performance levels can be determined with more certainty than future taskings, and that lack of proximity mitigates severity. However, an outcome where an increasing number of shortfalls occurs as time progresses would be undesirable because an unacceptably high amount of tasking shortfall may occur within a relatively short amount of time near the end of the horizon. For each year in the planning horizon, we measure the deviation from required tasking, *obtained from models for the corresponding trial cases without the confounding effects of cost discounting* (Trials X.3). We compare each amount of shortfall against the original case. In the absence of cost discounting, the models are more difficult to solve; hence, we use the best solutions found in a "reasonable" amount of time, i.e., no more than several hours. In many instances, we actually find that the taskings are met to a greater extent *with* the presence of the discounting factor for the shortfall penalty.

Table 4 lists relative tasking shortfalls for selected trials as the ratio of the sum of tasking shortfalls in the original model to the related model with penalty discounting. Ratios are given for each year in the planning horizon. A ratio greater than 1 indicates that the trial meets the tasking requirements to a greater extent than the original model. We suggest that the presence of the discounting factor helps the solver distinguish among many similar solutions; the improvement in meeting tasking requirements is actually a function of a better solution (i.e., a solution with a lower gap). This is best illustrated by Trial 3a.3, which differs from the original model only in branching hierarchy and the imposition of the penalty discounting. The former solution possesses a 5.1% gap whereas the best solution obtainable from the latter case possesses a gap of 19.4%. Conversely, for those cases in which the absence of cost discounting results in a relatively poor solution, taskings are not met as well overall as in the original model. Trial 6.3 with a 50% gap provides the best example of this.

Ratio of Original to Trial tasking shortfall (XDEV) Values for Each Year (and gap)				
Year	Trial 3a.3 (5.1%)	Trial 7.3 (5.6%)	Trial 2.3 (15%)	Trial 6.3 (50%)
1998	1.000	1.000	1.000	1.000
1999	1.000	1.000	1.000	1.000
2000	1.002	0.989	1.000	0.965
2001	0.995	0.974	1.012	0.934
2002	0.995	0.997	1.012	0.970
2003	0.995	0.997	1.012	1.066
2004	1.000	0.973	0.977	0.865
2005	1.000	0.973	0.977	0.904
2006	1.007	0.996	0.994	0.819
2007	1.007	0.996	0.994	1.116
2008	1.016	0.984	0.960	0.949
2009	1.016	0.984	0.960	0.949
2010	1.057	1.087	0.985	0.970
2011	1.057	1.082	0.985	0.970
2012	1.009	0.999	0.955	0.880
2013	1.021	1.012	0.967	0.922
2014	1.038	1.028	0.984	0.950
2015	1.038	1.024	0.984	0.950
2016	1.044	1.029	0.983	0.933
2017	1.044	1.029	0.983	0.771
2018	1.056	1.069	1.021	0.801
2019	1.056	1.069	1.021	0.783
2020	1.047	1.060	1.008	0.774
2021	1.047	1.060	1.008	0.732
2022	1.065	1.094	0.891	0.690
2023	1.065	1.105	0.861	0.552

Table 4: Comparisons between the original case and selected trial cases where we only discount penalties for the shortfall in task performance. Given are ratios of the sum of deviations from performance standard for each year in the planning horizon, i.e., a ratio greater than 1 indicates that the trial meets tasking requirements to a greater extent than the original model. For example, deviations from tasking requirements are the same for the original model and Trial 3a.3 for 1998 and 1999. Tasking requirements are met to a greater extent with the original model for years 2001-2003. In all other time periods, Trial 3a.3 matches or outperforms the original model – most significantly, in the last few years. Gaps reported and solutions used reflect the best obtainable in a "reasonable" amount of time.

We also note that both annual and epochal budget over-expenditures occur in the solution of the original model, whereas they are absent from the trial cases in which the gaps are about 5%. As expected, the poorer-quality solutions (Trials 2.3 and 6.3) yield over-expenditures, the greater of which occurs in the solution with the higher gap (Trial 6.3). These numerical results appear in Table 5. Therefore, the addition of the discounting factor provides relatively good solutions with respect to budget overspending. The overspending that does occur could be mitigated by a variety of modeling alternatives, for example, with the addition to the objective

function of a piecewise-linear penalty term (increasing the penalty with the amount of overexpenditure).

	Millions of Dollars over Budget in the Given Year (or interval)									
	Trial 1.2		Trial 3a.3		Trial 7.3		Trial 2.3		Trial 6.3	
Year	***bover***	***lover***	***bover***	***lover***	***bover***	***lover***	***bover***	***lover***	***bover***	***lover***
2001	133.5	0	0	0	0	0	77.1	0	0	0
2002	0	0	0	0	0	0	0	0	19.2	0
2003	0	0	0	0	0	0	67.6	0	0	0
2007	0	0	0	0	0	0	0	0	0	654.5
2008	18.2	0	0	0	0	0	0	0	0	0
2011	0	0	0	0	0	0	12.6	0	0	0
2012	0	0	0	0	0	0	0	67.6	0	0
2017	0	274.9	0	0	0	0	0	70.7	0	517.2
2022	0	758.9	0	0	0	0	0	0	0	1007.6

Table 5: Budget over-expenditures for the original model and selected trial cases where we only discount penalties for the shortfall in task performance. For example, in Trial 1.2, budgets excluding launch costs are exceeded in 2001 and 2008 by $133.5 and $18.2 million, respectively; budgets considering launch requirements are exceed by $274.9 and $758.9 million in 2017 and 2022, respectively. Gaps reported and solutions used reflect the best obtainable in a "reasonable" amount of time.

We must also evaluate the quality of solutions with the introduction of the continuous-valued research-and-development concepts. By the criteria discussed in Section IV.A, about 80% of the systems qualify as research-and-development concepts, and of the approximately 15,200 valid start-stop combinations, more than 12,500 correspond to those associated with research-and-development projects. Allowing a large number of fractional concepts could result in an unimplementable solution if any of the following materialized: (i) an inordinate number of fractional concepts, (ii) a large number of concepts with funding streams split into small allocations many times over the horizon, or (iii) many fractionally-funded research-and-development concepts receiving only a small proportion of their total budget.

We find none of these to be the case. Table 6 demonstrates that in the four relevant trials, we find no more than 12 fractionally-funded research-and-development concepts even including Trial 8.2, which exhibits fairly poor computational performance. Overall, fractionally-funded concepts account for no more than 20% of the total number of funded systems.

	Number of Systems Funded			
Trial	**Non- R&D**	**R&D: integral**	**R&D: fractional**	**% fractional systems**
5.2	32	23	9	14%
5a.2	41	23	8	11%
7.2	32	17	12	20%
8.2	27	19	11	19%

Table 6: Relative numbers of fractional and integral systems funded. For example, of the 64 systems funded in Trial 5.2, 32 are non-research-and-development concepts, 23 are research-and-development (R&D) concepts that are allocated full funding in one allotment, and 9 receive fractional funding. These nine concepts constitute only 14% of the total number of funded systems. In all cases, at most one-fifth of the systems are fractionally funded.

Additionally, we note that at least half of the concepts in each trial receive funding only once throughout the horizon (with the exception of Trial 5a.2); in these instances, there is no change to the nature of the funding stream, but rather to the total amount of money allocated to a concept. Of those concepts that are funded multiple times over the horizon, the vast majority of these are provided with partial funding no more than twice throughout the horizon. These results are displayed in Table 7.

Number of Research-and-development Concepts With the Following Number of "Allocations"					
Trial	**1**	**2**	**3**	**4**	**Total**
5.2	5	3	1	0	9
5a.2	0	5	1	2	8
7.2	6	5	1	0	12
8.2	8	3	0	0	11

Table 7: Only a modest number of research-and-development concepts are allocated fractional funding, and most of these receive partial funding only once throughout the horizon. Of this modest number, there are a few instances of a concept receiving funding in three or even four allotments. For example, five fractionally-funded concepts in Trial 5.2 receive funding only once during the horizon; three receive funding twice, and one concept receives three funding allotments.

Finally, in two of the three cases in which concepts are not required to receive full funding over the horizon, over 65% of them are allocated at least 50% of their funding. In the third case, Trial 8.2, more than half of them are allocated at least 50% of the requested funding. Most concepts receive full funding, though sometimes through two or more allotments over the horizon. We present numerical results in Table 8.

Number of Research-and-development Concepts With the Following Percentage of Funding					
Trial	**100%**	**80-99%**	**50-79%**	**20-49%**	**<20%**
5.2	3	0	3	0	3
7.2	6	0	4	0	2
8.2	5	0	1	3	2

Table 8: Of the research-and-development concepts receiving fractional funding, for the trials we tested, more than half receive at least 50% of their desired allocation. In a few instances, concepts receive full funding divided into two or more allotments over the horizon. For example, of the nine fractional concepts receiving funding in Trial 5.2, three of them receive full funding (in several allotments), three receive between 50% and 79% of their funding, and three concepts receive less than 20% of their required funding.

Allowing the integer variables representing research-and-development concepts to be relaxed yields good-quality solutions quickly. Furthermore, only a scant number of the research-and-development concepts assume fractional values. Most of the fractionally-funded concepts receive a significant portion of their funding in few allotments. However, this outcome cannot be guaranteed. Additionally, the utility of a fractional concept is assumed to yield a linear contribution proportional to its fractional funding allotment. It may be unreasonable to assume that a concept with half its funding can contribute half the utility of the same fully-funded concept, or, perhaps, that a concept funded in this manner would become fully operationally capable after the same amount of time as a concept with integral funding would. The Air Force Materiel Command Office of Aerospace Studies must judge whether this shortcoming is acceptable, possibly changing the set of research-and-development concepts accordingly.

VI. CONCLUSION

SCOUT is a mixed-integer linear program that recommends a mix of current systems, future concepts, and launches to minimize shortfalls in task performance while adhering to constraints on budget, launch vehicle demand, launch vehicle availability, and logic governing the precedence and interdependence of systems. Model instances contain about ten thousand constraints, several thousand continuous variables, and an equal number of binary variables. Our results suggest that SCOUT's computational requirements can be significantly reduced (from hours to minutes) and SCOUT's integrality gaps can be tightened by applying discount factors to both costs and tasking shortfall penalties, and by using continuous variables to model some research-and-development concepts. These two modifications result in the most improvement in performance throughout all computational experiments without significantly altering solution quality. Reducing time-period granularity results in only modest performance enhancement. The branching hierarchies we discuss have an unpredictable influence on results. The effects of combinations of modifications we impose simultaneously are inconclusive. We encourage the introduction of discounting and fractional research-and-development concepts, keeping in mind that the greatest achievements in performance often occur within the first ten to twenty minutes of run time.

ACKNOWLEDGEMENTS

The Naval Postgraduate School authors thank USAF LtCol Kirk Yost for introducing us to SCOUT, and for his early improvements to the model. We acknowledge the helpful comments received from several people, including USAF LT Heath Holtz and Dr. Joe Melton, associated with the Office of Aerospace Studies, Kirtland AFB, NM on earlier drafts of this report. We thank Naval Postgraduate School Professor Kevin Wood for his improvements on the last versions of this report. This research was funded by the United States Air Force Space Command.

VII. REFERENCES

Anderson, Michael R., TRADOC Analysis Center. Personal communication, January 22, 1999.

Brooke, A., Kendrick, D., Meeraus A., and R. Raman. GAMS: A User's Guide. The Scientific Press. 1997.

Brown, Gerald G., Clemence, Robert D, Teufert, William R. and R. Kevin Wood. "An Optimization Model for Modernizing the Army's Helicopter Fleet." *Interfaces*. Vol. 12, No. 4, 1991, pp. 39-52.

Brown, Gerald G., Dell, Robert F. and R. Kevin Wood. " Optimization and Persistence." *Interfaces*. Vol. 27, No. 5, 1997, pp. 15-37.

Carr, Douglas A. "Optimally Scheduling Theater Missile Defense Procurement." MS Thesis in Operations Research, Naval Postgraduate School, Monterey, CA, 1996.

Clark, J. J., Hindelang, T. J. and R. E. Pritchard. Capital-budgeting: Planning and Control of Capital Expenditures. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

Donahue, Scott F. "An Optimization Model for Army Planning and Programming." MS Thesis in Operations Research, Naval Postgraduate School, Monterey, CA, 1992.

Gooley, Timothy (Major, US Air Force). Personal communication with Lt. Col. Kevin Keefer (USAF), August 6, 1998.

Gross, William M. "Integer Programming Based Investment Trade-off." MS Thesis in Operations Research, Naval Postgraduate School, Monterey, CA, 1996.

Hummeltenberg, W. "Capital-budgeting with Benders Decomposition." *European Journal of Operational Research*. Vol. 21, No. 3, 1985, pp. 318-329.

Ihde, Alexander G. "An Optimization Model for Anti-Armor Weapon Systems Acquisition." MS Thesis in Operations Research, Naval Postgraduate School, Monterey, CA, 1995.

Iwamura, K. and B. Liu. "Chance Constrained Integer Programming Models for Capital-budgeting in Fuzzy Environments." *Journal of the Operational Research Society*. Vol. 49, No. 8, 1998, pp. 854-860.

Karabakal, Nejat, Lohmann, Jack R. and James C. Bean. "Parallel Replacement under Capital Rationing Constraints." *Management Science*. Vol. 40, No. 3, 1994, pp. 305-319.

Keown, Arthur J. and Bernard W. Taylor III. "A Chance Constrained Integer Goal Programming Model for Capital-budgeting in the Production Area." *Journal of the Operational Research Society*. Vol. 31, No. 7, 1980, pp. 579-589.

Kumar, P. C. and Trami Lu. "Capital-budgeting Decisions in Large Scale, Integrated Projects: Case Study of a Mathematical Programming Application." *The Engineering Economist*. Vol. 36, No. 2, 1991, pp. 127-150.

Lotfi, V., Sarkis, J. and J. H. Semple. "Economic Justification for Incremental Implementation of Advanced Manufacturing Systems." *Journal of the Operational Research Society*. Vol. 49, No. 8, 1998, pp. 829-839.

Mamer, John W. and Andrew W. Shogan. "A Constrained Capital-budgeting Problem with Applications to Repair Kit Selection." *Management Science*. Vol. 33, No. 6, 1987, pp. 800-806.

Parnell, Gregory S., Conley, Harry W., Jackson, Jack A., Lehmkuhl, Lee J. and John M. Andrew. "Foundations 2025: A Value Model for Evaluating Future Air and Space Forces." *Management Science*. Vol. 44, No. 10, 1998, pp. 1336-1350.

Taylor, Bernard W. III, Keown, Arthur J. and Allen G. Greenwood. "An Integer Goal Programming Model for Determining Military Aircraft Expenditures." *Journal of the Operational Research Society*. Vol. 34, No. 5, 1983, pp. 379-390.

United States Air Force. "Air Force Titan IV Launch Successful." http://www.af.mil. Accessed March 10, 1999a.

United States Air Force. "X-33 Space Plane to Call Edwards Home." http://www.af.mil. Accessed March 10, 1999b.

Weingartner, H. M. Mathematical Programming and the Analysis of Capital-budgeting Problems. Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

APPENDIX

Tasks are components of *capabilities*, which, in turn, are components of *planning areas*. Following is a list of all of these, as used by the Air Force Space Command.

Planning Area 1: Aerospace superiority
- Capability 1.1: Dominate operations in the air medium
  - Task 1.1.1: Suppress adversary air defenses
  - Task 1.1.2: Neutralize adversary air and cruise missile capabilities
- Capability 1.2: Dominate operations in, from, and through the space medium
  - Task 1.2.1: Protect friendly space capabilities
  - Task 1.2.2: Protect friendly missile capabilities
  - Task 1.2.3: Neutralize adversary space capabilities
  - Task 1.2.4: Provide national and theater missile defense
  - Task 1.2.5: Operate space assets

Planning Area 2: Rapid global mobility
- Capability 2.1: Provide access to, from, and through space
  - Task 2.1.1: Deploy space assets

Planning Area 3: Global attack and precision engagement
- Capability 3.1: Perform global attack of surface targets
  - Task 3.1.1: Neutralize non-weapons-of-mass-destruction targets
  - Task 3.1.2: Deter and counter weapons of mass destruction

Planning Area 4: Information superiority
- Capability 4.1: Dominate operations in the infosphere
  - Task 4.1.1: Conduct defensive counter information operations
  - Task 4.1.2: Conduct offensive counter information operations
- Capability 4.2: Gain and exploit information
  - Task 4.2.1: Provide information on space events, activities, and threats
  - Task 4.2.2: Provide information on air events, activities, and threats
  - Task 4.2.3: Provide information on surface and sub-surface events, activities, and threats
  - Task 4.2.4: Provide information on infosphere events, activities, and threats
  - Task 4.2.5: Provide environmental monitoring information
  - Task 4.2.6: Provide global satellite communications
  - Task 4.2.7: Provide global positioning and timing information

Planning Area 5: Global Awareness and Command and Control
- Capability 5.1: Provide command and control
  - Task 5.1.1: Monitor global conditions and events
  - Task 5.1.2: Assess global conditions and events
  - Task 5.1.3: Plan military operations
  - Task 5.1.4: Execute military operations

Planning Area 6: Agile combat support
- Capability 6.1: Protect forces
  - Task 6.1.1: Provide physical protection of ground assets
- Capability 6.2: Sustain forces
  - Task 6.2.1: Provide weapon system life cycle maintenance and sustainment
  - Task 6.2.2: Conduct force development evaluations of space and missile forces
  - Task 6.2.3: Conduct readiness exercises
- Capability 6.3: Support Installations
  - Task 6.3.1: Develop, operate and maintain facilities
  - Task 6.3.2: Provide contingency engineering support
  - Task 6.3.3: Provide transportation support

Planning Area 7: Quality people
- Capability 7.1: Train personnel
  - Task 7.1.1: Provide operational training
- Capability 7.2: Educate personnel
- Capability 7.3: Retain personnel
  - Task 7.3.1: Provide quality of life
- Capability 7.4: Access (recruit) personnel
- Capability 7.5: Promote good order and discipline

Planning Area 8: Innovation
- Capability 8.1: Provide Innovation
  - Task 8.1.1: Support innovation

INITIAL DISTRIBUTION LIST

1. Research Office (Code 09) 1
   Naval Postgraduate School
   Monterey, CA 93943-5000

2. Dudley Knox Library (Code 013) 2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

4. Therese Bilodeau (Editorial Assistant) 1
   Dept of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Alexandra M. Newman (Code OR/Nl) 5
   Dept of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. Gerald G. Brown (Code OR/Bw) 5
   Dept of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

7. Robert F. Dell (Code OR/De) 5
   Dept of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

8. Richard E. Rosenthal (Code OR/Rl) 5
   Dept of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

9. Angela Giddings 1
   316 Jersey Lane
   Lafayette, IN 47909

10. Heath Holtz 3
    AFMC OAS/DRA
    3550 Aberdeen Ave. SE
    Kirtland AFB, NM 87117-5776

11. Joe Melton 3
    SWC/AE
    730 Irwin Ave., Suite 83
    Schriever AFB, CO 80912-7383